

BAB II

LANDASAN TEORI

Pada bab ini akan merengkan tentang landasan teori serta hasil penelitian terkait yang dapat mendukung serta mendasari penelitian ini. Tujuan dari pembuatan dasar teori ini untuk membantu proses penulisan pada bab selanjutnya.

2.1 Hasil Penelitian Terkait

Aplikasi Stok dan Transaksi Local Database server Berbasis Web : Pada took Metromas”. Pada penelitian tersebut menghasilkan sebuah aplikasi yang dapat menampung aplikasi menggunakan bahasa pemrograman PHP, dalam aplikasi ini disediakan menu customer, transaksi pembelian dan admin[6].

Kemudian Penelitian dengan judul “Pengembangan Aplikasi Database Persediaan Barang Berbasis Web pada wahana Komputer Semarang”. Pada penelitian ini menghasilkan sebuah aplikasi yang diperuntukan untuk bagian-bagian tertentu seperti pimpinan, dan bagian stok[7].

2.2 Web Server

Mengenai server sebagai sebuah pusat, dan juga difungsikan sebagai pelayan yang berguna untuk pengiriman dan juga penerimaan data, serta mengatur proses pengiriman dan juga penerimaan data diantara komputer-komputer yang tersambung, atau dengan kata lain, server memiliki fungsi utama sebagai penyedia sebuah layanan bagi client atau user.

Macam-macam webs server di antaranya[8] :

- a. Apache Web Server / The HTTP Web Server
- b. Apache Tomcat
- c. Microsoft Windows Server 2008 (Internet Information Service)
- d. Lighttpd
- e. Zeus Web Server
- f. Sun Java System Web Server

2.3 User Interface (UI)

Antarmuka pengguna atau *user interface* (UI) adalah suatu istilah yang digunakan untuk menggambarkan tampilan dari mesin atau komputer yang berinteraksi langsung dengan pengguna. Desain dan penyusunan tampilan antarmuka perlu diperhatikan untuk menghasilkan tampilan yang bagus. Ada beberapa metode yang digunakan dalam menggambarkan *user interface* yaitu metode *Flat Design*, metode *User Centered Design*, *Design Guidelines and Standards*.

Flat Design adalah genre (aliran) desain UI minimalis, atau bahasa desain yang sedang banyak digunakan oleh *user interface grapher* (seperti *website* dan *web apps*). *Flat design* mengacu kepada *style desain interface* yang menghilangkan pilihan *stylistic* yang memberikan ilusi 3 dimensional (seperti *drop shadow*, *gradient*, *texture*, dan *depth*) dan fokus terhadap penggunaan elemen minimalis, seperti tipografi dan *flat colors* (warna datar).

2.4 PhpMyAdmin

PhpMyAdmin adalah aplikasi berbasis web yang dibuat dari pemrograman PHP dan diramu dengan JavaScript. phpMyAdmin juga dapat disebut sebagai tools yang berguna untuk mengakses ada database MySQL Server dalam bentuk tampilan web. Dengan adanya phpMyAdmin semua pekerjaan menjadi lebih mudah, karena tanpa harus mengerti perintah-perintah dasar SQL, kita sudah dapat memanajemen database dan data di dalamnya [10].



Gambar 2.1 Logo PhpMyadmin

2.5 MySQL (My Structured Query Language)

MySQL adalah database yang menghubungkan script PHP menggunakan perintah query dan escape character yang sama dengan PHP. PHP memang mendukung banyak jenis database, tetapi untuk membuat web yang dinamis dan selau up to date, MySQL merupakan pilihan database tercepat saat ini[12].



Gambar 2.2 Logo MySQL (My Structured Query Language)SS

2.6 Xampp

XAMPP merupakan paket PHP dan MySQL berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP. XAMPP mengkombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket[10].



Gambar 2.3 Logo Xampp

2.7 CodeIgniter

CodeIgniter (CI) adalah framework perkembangan aplikasi dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis[11].



Gambar 2.4 Logo CodeIgniter

2.8 Konsep Tentang Unfied Modeling Language (UML)




UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modelling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami[9].







a. Use Case Diagram

Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan.

Tidak ada cara standar penulisan isi sebuah use case dan format-format yang berbeda dapat bekerja dengan baik dalam kasus yang berbeda. Setiap use case memiliki aktor utama yang meminta sistem untuk memberi sebuah layanan[12].

Tabel 2.1 *Simbol Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri(<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .






5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. Activity Diagram

Activity Diagram adalah teknik menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memerankan peran mirip sebuah diagram alir, tetapi

perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung behavior paralel. Activity diagram telah mengalami beberapa perubahan paling besar selama perkembangan versi-versi UML, jadi tidaklah mengejutkan jika activity diagram telah dikembangkan secara signifikan dan diubah lagi dalam UML 2. Dalam UML 1, activity diagram dianggap sebagai kasus khusus state diagram [12].

Tabel. 2.2 Simbol *Activity Diagram*


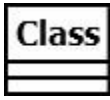


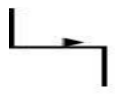
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

c. *Class Diagram*

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara mereka. Class diagram juga menunjukkan properti dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah class. Class diagram tidak hanya digunakan secara luas tetapi juga memiliki

banyak konsep pemodelan. Meskipun elemen-elemen dasar dibutuhkan oleh setiap orang, diagram dengan spesifikasi kegunaan untuk mengkomunikasikan visualisasi interaksi antar kelas dalam sistem [12].




Tabel 2.3 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Generalization	Hubungan dimana obyek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor)
2		Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
3		Realization	Operasi yang benar-benar dilakukan oleh suatu objek
4		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
5		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya

d. *Sequence Diagram*

Interaction diagram menunjukkan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa behavior. UML memiliki beberapa bentuk *interaction* diagram dan yang paling umum digunakan adalah sequence diagram, secara khusus, menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek didalam use case[12].

Tabel 2.4 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		LifeLine	Objek entity, antar muka yang saling berinteraksi
2		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

e. Class Responsibility Collaborator (CRC)

CRC (Class Responsibility Collaborator) adalah kumpulan kartu indeks standard yang telah dibagi menjadi tiga (class, responsibility, collaborator). Layout dari CRC adalah sebagai berikut :

Tabel 2.5 Simbol *CRC*

Class Name	
Responsibility	Collaborator